

525088

<b>Title</b>	Method and apparatus for generating structured documents for various presentations
<b>Publication Date</b>	2003/03/21
<b>Certification_Number</b>	174819
<b>Application Date</b>	2001/09/28
<b>Application No.</b>	090124215
<b>IPC</b>	<b>G06F-009/44</b>
<b>Inventor</b>	HUANG, EVAN S. US
<b>Applicant</b>	XML CITIES, INC. US
<b>Priority Number</b>	2001/01/05 US20010754969

**Abstract**

The use of identifiers in user-defined document type definitions is disclosed for converting unstructured documents to structured documents. The identifiers in user-defined document type definitions are used to associate selected objects or group objects in the unstructured documents so that association information of the selected objects or group objects can facilitate the generation of files in a markup language suitable for presentations on various media.

# 公告本

申請日期	90. 9. 28
案 號	90124215
類 別	G06F 9/44

A4  
C4

525088

(以上各欄由本局填註)

## 發明專利說明書

一、發明名稱	中 文	產生各種表現方式之結構化文件的方法及裝置
	英 文	METHOD AND APPARATUS FOR GENERATING STRUCTURED DOCUMENTS FOR VARIOUS PRESENTATIONS
二、發明人	姓 名	黃艾凡 S.
	國 籍	美國
	住、居所	美國加州卡伯堤諾香橙花大道 7634 號
三、申請人	姓 名 (名稱)	美商・XML 都市股份有限公司
	國 籍	美國
	住、居所 (事務所)	美國加州梅爾皮塔斯南大街 1617A 號
	代 表 人 姓 名	黃文生

第 1 頁

本紙張尺度適用中國國家標準 (CNS) A4 規格 (210×297 公釐)

(由本局填寫)

承辦人代碼：
大 類：
I P C 分類：

A6  
B6

本案已向：

國 ( 地區 )    申請專利，申請日期：                      案號：                      ， ☒ 有    ☐ 無主張優先權

本案已向美國申請專利；申請日：2001 年 1 月 5 日    案號：09/754,969 號

有關微生物已寄存於：                                      ，寄存日期：                                      ，寄存號碼：

(請先閱讀背面之注意事項再填寫本頁各欄)

裝

訂

線

經濟部智慧財產局員工消費合作社印製

## 四、中文發明摘要(發明之名稱：)

產生各種表現方式之結構化文件的方法及裝置

茲揭示一種使用者定義之文件型態定義內的識別子應用方法，以將未結構化文件轉換為結構化文件。利用該等使用者定義之文件型態定義內的識別子，將未結構化文件內的選定物件或物件群組相予關聯，令該等選定物件或物件群組之相關性資訊助於產生適合在各種媒體上表現的標記語言檔案。

## 英文發明摘要(發明之名稱：)

**Method and Apparatus for Generating Structured Documents for Various Presentations**

The use of identifiers in user-defined document type definitions is disclosed for converting unstructured documents to structured documents. The identifiers in user-defined document type definitions are used to associate selected objects or group objects in the unstructured documents so that association information of the selected objects or group objects can facilitate the generation of files in a markup language suitable for presentations on various media.

## 五、發明說明( )

### 相關申請案：

本申請案已於西元 2000 年 1 月 31 日申辦臨時專利申請案第 60/179,330 號，標題為“Method and Apparatus for Generating Structured Documents with User-defined Document Type Definitions Using Structure-based Font Information”，茲將其併入本案，以作為參考。

### 發明背景：

本發明概有關於文件處理及電子出版系統範疇，且尤關於一種用以按使用者定義之文件型態定義來產生結構化文件的方法及裝置。本發明也有關於一種機制，可讓使用者利用本方法及裝置按各種表現方式來轉換未結構化文件，其中將該未結構化文件定義成透過一編修應用程式(即如文書處理)所組合、編輯或管理的各種檔案。

### 相關技藝說明：

網際網路係一快速成長的全球互連電腦通訊網路。而數以百萬計的互連電腦一起構成一個龐大的超鏈結資訊存庫，且即可由世界各地任何時間的所有連線電腦接取。由於既已產生出成千上萬的網頁且加入這個龐大存庫裡，因此會產生一種重大需求，俾以將像是簡報、試算表或簡冊等文件，快速且簡易地轉換成為一種可由網際網路上其他應用程式或電腦所表現的格式。

現已眾知對於網頁瀏覽應用程式(如瀏覽器)表現之較

## 五、發明說明( )

適格式係按標記語言，如“超文字標記語言(HTML)”、“可延展標記語言(XML)”、“標準通用標記語言(SGML)”或是“無線標記語言(WML)”等。依此方式所組成、編輯或管理之網頁瀏覽應用程式的檔案或文件可泛稱為結構化檔案或文件。在所有的結構化文件益處中，可提供使用者定義之文件型態定義(DTD)或文件法則定義的能力，設立了新的資訊交換或儲存典範。然而，其挑戰之處在於如何按使用者所自行定義的 DTD 來產生結構化文件。

可要不從未結構化文件，要不轉換自具有其他型態之 DTD 的結構化文件，來產生一個具有特定 DTD 的未結構化文件。現有多種用以產生結構化文件的編輯器。這些示範性編輯器包括 Adobe FrameMaker、Arbortext Epic 及 SoftQuad Xmetal。這些編輯器通常提供一結構化視窗，連帶一文書處理視窗，在此該文書處理視窗即如未結構化文件所用的傳統式文書處理環境，而該結構化視窗含有定義於某 DTD 內之各資料單元的文件結構。為要在這些編輯器內從文稿中產生出結構化文件，使用者一般會需要在文書處理視窗內產生一未結構化文件。當載入所欲的 DTD 後，該使用者即根據該 DTD 內所定義的各文件單元，而在該結構視窗中構築出一文件結構樹狀物。通常，使用者會藉從所產生的文件將各文件單元拷貼或拖放到該文件結構樹狀物來進行各項處理程序。

為於這些編輯器裡將具有一 DTD 的結構化文件轉換

## 五、發明說明( )

成另一 DTD，使用者需要載入該結構化文件，將各文件單元的標籤及屬性從一 DTD 修改成另一者，然後混合各文件單元或是剖析與新 DTD 內重新定義之文件單元相關聯的新文件單元。

在眾多如前所述的程序中，各資料單元與文件單元之間的關聯性，對於產生或轉換一未結構化或結構化文件成為一具有特定 DTD 之結構化文件來說確屬關鍵且費力的處理作業。如今既已提出多種方法以將資料單元和文件單元相關聯，俾簡化該結構化文件的產生作業。例如，關鍵字擷取方法可自未結構化文件中擷取出文件結構的關鍵字表現形式，並且利用關鍵字/文字組對來作為各文件單元與資料單元之間的關聯性。一種等位方法可藉由對各等位文件的座標加以排序，將各資料單元與各文件單元內的標記語言標籤相互關聯。一種邏輯結構方法可藉由比對預設樣式以分析該文件結構，並根據所分析的各文件單元來剖析各資料單元。然上述各種方法無一考慮到利用識別子(即如字型資訊)來將各資料單元與文件單元相互關聯。因此，即需要一種通用方法，可利用使用者定義之文件型態定義內的識別子資訊，將各資料單元與文件單元相互關聯以產生結構化文件。

此外，各示範性編輯器所要求的各項程序亦顯冗瑣且煩耗，同時意味著內含的高成本。經常，某個擁有大量文件待加轉換的企業因該項轉換處理缺乏效率且費時遲緩之故，從而必須要將其委外處理。另一方面，由一服務供

## 五、發明說明( )

應商所施作的這項轉換處理亦不易加以量化，因為這主要是牽涉到依據各文件複雜度而定的人工及重覆性處理作業。從而即另需要一種機制，可按照成本算定的方式，將未結構化文件轉成對於各種表現方式之結構化文件的轉換作業予以量化。

### 發明目的及概述：

本發明係按慮及前述各項問題及需要而製作，並對於在網際網路上各種表現方式具有特定的應用性。本發明之一特點在於利用 DTD 檔案內的識別子以關聯選定物件或物件群組，使得所選定之物件或物件群組的相關性資訊協助產生適合於在各種媒體上表現方式的標記語言檔案。

本發明可按方法、系統、產品或其他實際形式所實作。根據其一具體實施例，本發明可為一方法。該方法接收一含有文件型態定義(DTD)的定義檔案，並將一超檔案且連同該定義檔案予以顯示，其中該超檔案含有數個可顯示物件，及其關於該等可顯示物件各者的各個裝飾屬性。該定義檔案包括各文件單元的結構，各者對應於該超檔案內各可顯示物件之一者。部分的文件單元包含數個識別子，這些識別子各個係被指配給該等文件單元其一者。在一實作中，這些識別子係屬數字及/或字母。而在其他實作中，這些識別子可為字型名稱、色彩名稱、大小、字型款式、色彩、式樣、各種效果或其他符號其中之一或多者。本方法可將至少一識別子相關於該等可顯示物件其中一



## 五、發明說明( )

者。

根據另一具體實施例，本發明可按一提供文件轉換處理之方法所實作，該方法包括啟動一具有編數系統之計數器、將一未結構化文件轉換成為超檔案，其中該超檔案含有數個可顯示物件，及關於該等可顯示物件各者的各裝飾屬性、接收包含關於該未結構化文件之文件型態定義(DTD)的定義檔案；產生一經修改之超檔案，該超檔案含有與該定義檔案內各定義其一相關之可顯示物件至少一者的相關性資訊；以及當需儲存該經修改超檔案時，即行引動該計數器以增量。

按後陳敘述以實作本發明，即可達到目的及優點並連同前揭說明，且獲致如隨附圖式所繪列之各款具體實施例。

### 圖式簡單說明：

參酌於後載說明、附纂申請專利範圍與隨示圖式，確可更為明瞭本發明前揭各項功能、特性、及優點，其中第 1A 圖顯示一基本系統組態，其內可根據一較佳具體實施例而實作本發明；

第 1B 圖顯示一系統的內部建構區塊，其內可實作且執行本發明，以獲得本發明考量之所欲結果；

第 2A 圖為一未結構化文件範例，可藉一編修工具加以組合、編輯或管理；

第 2B 圖為一文件型態定義(DTD)之範例；

## 五、發明說明( )

第 2C 圖顯示如圖 2A 所示之未結構化文件，依照圖 2B 所示之文件型態定義(DTD)的結構化文件；

第 3A 圖為根據本發明一具體實施例之功能圖；

第 3B 圖顯示實作出根據本發明一具體實施例之轉換模組的視像環境；

第 3C 圖顯示一根據第 3B 圖所示超檔案內之可顯示物件，而按 XML 格式所設計的文稿式樣範例；

第 3D 圖為一發自 Dongle 的示範性訊息；

第 3E 圖顯示一利用含有根據本發明一具體實施例之轉換模組實作產品的方法流程圖；

第 4 圖敘述一資料處理裝置區塊圖，該者可利用結構基礎式字型態資訊並按使用者定義之 DTD，輸入、編輯及轉換未結構化或結構化文件而將其轉為結構化文件；

第 5 圖為一如第 2 圖 DTD 中及字型屬性所定義之文件單元的相關性表單；

第 6 圖顯示對如第 1 圖未結構化文件的編輯結果，在此各個經剖析之資料單元既已根據第 5 圖內的相關性而被指配字型屬性；

第 7 圖顯示一轉換處理方法，可將第 6 圖內的經剖析資料單元，按第 2B 圖內示範性 DTD 轉為所欲之結構化文件；

第 8 圖為中介性結構化文件，該者含有剖析資料單元，及與一字型文件單元之列表相關的指配字型 ID；

(請先閱讀背面之注意事項再填寫本頁)

五、發明說明( )

第 9 圖顯示一轉換處理方法，可將第 6 圖內的中介性結構化文件，按第 2 圖內 DTD 轉為所欲之結構化文件；以及

第 10 圖顯示利用轉換用可延展式樣語言(XSLT)的第 9 圖轉換處理方法之實作結果。

圖號對照說明：

- |                        |                 |
|------------------------|-----------------|
| 100 電腦                 | 102 計算裝置        |
| 104 服務伺服器              | 106 桌上型電腦       |
| 108 資料網路               | 110 私屬網路        |
| 120 資料匯流排              | 122 中央處理單元(CPU) |
| 124 裝置介面               | 126 顯示介面        |
| 128 網路介面               | 130 印表機介面       |
| 132 主記憶體               | 134 唯讀記憶體       |
| 136 儲存裝置               | 138 軟碟機驅動介面     |
| 140 鍵盤                 | 142 點指裝置        |
| 200 未結構化文件             | 208 文件型態定義      |
| 210 文件單元               | 302 轉換模組        |
| 302 關聯模組               | 306 整合模組        |
| 308 計數器                | 310 經修改超檔案      |
| 320 環境                 | 322 顯示畫面        |
| 324 顯示畫面               | 326 群組物件        |
| 328 DTD Pool(文件型態定義儲庫) |                 |
| 330 XML 樹狀物            | 332 文件單元“標題”    |

(請先閱讀背面之注意事項再填寫本頁)

訂

線

五、發明說明( )

- |                            |            |
|----------------------------|------------|
| 334 “資料”                   | 400 功能性區塊圖 |
| 402 文件資料庫                  | 404 資料處理模組 |
| 406 輸入模組                   | 408 文件     |
| 410 編輯模組                   |            |
| 412 文件單元及字型屬性的相關性表單        |            |
| 414 轉換或過濾模組                |            |
| 416 根據字型屬性及文件單元的映對規則       |            |
| 418 未結構化/結構化文件、超檔案、文件資料庫   |            |
| 500 相關性表單                  | 502 文件單元   |
| 504 元件屬性                   | 506 字型款式   |
| 508 字型樣式                   | 510 字型色彩   |
| 512 字型大小                   | 514 字型效果   |
| 600 編輯結果                   | 602 物件     |
| 604 物件                     | 606 物件     |
| 608 物件                     | 610 物件     |
| 612 物件                     | 614 物件     |
| 802 含有具特定字型屬性之“字型”單元       |            |
| 804 含有具“font_ID”屬性的經剖析資料單元 |            |
| 808 中介結構化檔案                |            |

發明詳細說明：

在本發明後載詳細說明中設定有眾多特定細節，俾供通徹明瞭本發明。然對於熟諳本項技藝之人士而言，可實作出本發明而無須該等特定細節確屬顯而易見。在其他範

## 五、發明說明( )

例中，對於眾知方法、程序、元件及電路並未詳加說明，以避免非必要地模糊本發明特點。本詳細說明概按各程序、邏輯區塊、處理作業及其他符號表現方式所纂述，可直接或間接地代表出耦接於網路上之資料處理裝置的各項運作。這些處理說明及表現即為熟諳本項技藝人士採用以最為有效傳遞其等作業之本質予其他熟諳本項技藝人士的方式。

在此所稱「一具體實施例」或「某具體實施例」意指關聯於該具體實施例而述之特定功能、結構或特徵，會被包含在本發明至少一具體實施例之內。本規格說明全篇各處之語詞「在一具體實施例中」並非必然地意指相同的具體實施例，而個別或替代性具體實施例亦非彼此互斥於其他具體實施例。此外，表示出本發明其一或更多具體實施例之處理方法流程圖或圖式中的區塊順序，實並非內定地標示任何特定次序，亦不意味本發明內的任何限制。

現參考各份圖式，其中相仿參考編號係指全篇各觀視圖內之類似局部。第 1A 圖顯示一基本系統組態，其中可根據一較佳具體實施例而實作本發明。可利用執行於電腦 100 上的編修工具，來產生像是產品說明書、功能表單及價目表的未結構化文件。由編修工具所產生的檔案或文件會被稱為未結構化文件。示範性的編修工具可包含 Microsoft Office (即如 Microsoft Word、Microsoft PowerPoint 以及 Microsoft Excel)、Adobe FrameMaker 和 Adobe Photoshop。這些未結構化文件可被上載到作為中央

## 五、發明說明( )

存庫之計算裝置 102 處。該計算裝置 102 可為 Sun Microsystems (www.sun.com) 出售的伺服器站臺，或一裝載有編譯且鏈結版本而得實作本發明之具體實施例的桌上型電腦。

在一款設定方式中，該電腦 100 及該計算裝置 102 屬不可分割者，並且執行文件轉換處理而產生出可最終地按如 XML 或 HTML 標記語言之格式所表示的結構化文件。在一應用方式中，這些按 XML 所表示的結構化文件會被轉化成 HTML 格式，並且成為可供經私屬網路 110 而連到主導著概稱為 www (全球資訊網) 的站臺之服務伺服器 104 處所運用。

在一種情況下，使用者可利用操控瀏覽器應用程式且經耦接於資料網路 108 的桌上型電腦 106，來接取該服務伺服器 104 上各份檔案。這些由該計算裝置 102 內的結構化文件所表示之檔案代表經一編修工具所創組的最新產品資訊。

即如後文中所詳釋，本發明並不限於網際網路應用。該者亦可執作於其中使用者經常會產生按照像是 FrameMaker 或 Microsoft Word 等不同文書處理格式之文件的個別電腦上。本發明可被用來將文件轉換成一種標記表現方式，而無論該精確文書處理格式為何。

第 1B 圖顯示一可實作並執行本發明之系統 118 的內部建構區塊。該系統 118 可對應於一客戶端裝置(即如電腦 100、102 或 106)或一伺服器裝置(即如伺服器 104)。正

## 五、發明說明( )

如第 1B 圖所示，該系統 118 包括一中央處理單元(CPU) 122，該者介接於一資料匯流排 120 及一裝置介面 124。該 CPU 122

可執行一些指令以管理耦接於該資料匯流排 120 的所有裝置及介面俾供同步運作，且裝置介面 124 可被耦接於一像是電腦 102 的外部裝置，如是則由此所傳來的文件會經由該資料匯流排 120 而被收存到記憶體或儲存裝置內。並且，顯示介面 126、網路介面 128、印表機介面 130 與軟碟機驅動介面 138 也介接於該資料匯流排 120。通常，可透過軟碟機驅動介面 138、網路介面 128、裝置介面 124 或其他耦接於該資料匯流排 120 的介面，而將本發明一具體實施例之經編譯且鏈結版本載入該儲存裝置 136 內。

如隨機存取記憶體(RAM)的主記憶體 132 也介接於該資料匯流排 120，以提供該 CPU 122 各項指令，並可接取到該記憶體儲存裝置 136 以取得資料及其他指令。特別是，當執行所存放的應用程式指令時，像是本發明經編譯且鏈結版本，將會引動該 CPU 122 以操控資料而獲得本發明所慮及之結果。在此供置有 ROM (唯讀記憶體) 134 以存放不變性指令序列，即如基本輸入/輸出作業系統(BIOS) 以供鍵盤 140、顯示器 126 與(如有)點指裝置 142 運作。

第 2A 圖說明一個由一編修工具所組創、編輯或管理之未結構化文件 200 範例。在一未結構化文件中，通常是按序列方式所呈現，而這一般會跟隨在一讀取命令之後(即如從上而下，從左而右)。該項序列可被剖析成眾多資

## 五、發明說明( )

料單元的區段，其中各個資料單元 102 會被指配予裝飾屬性或資訊，像是位置、字型色彩、字型大小、字型款式、式樣及各種效果等等。該裝飾資訊的目的，基本上是為了當編修工具開啟含有資料單元之檔案以供顯示於一顯示螢幕上時，可獲適當的排置及表現。

根據一具體實施例，一未結構化文件會被印出為含有該裝飾資訊的超文件格式。該超文件格式的範例即為大家常用的“可攜式料格式(PDF)”。這種超文件格式的優點是其離於該編修工具且或可離於電腦的獨立性，使得可在各種環境下等同地開啟或讀取該超文件格式。

像是 SGML 與 XML 的結構化文件是以文件型態定義(DTD)為開始。第 2B 圖說明一“食譜型態”文件的 DTD 208 範例，其中會將文件打破成為許多文件單元結構。其一特定文件單元 210 可含有其他的文件單元及屬性。該文件單元的另一範例為僅含有經剖析的字元資料。

第 2C 圖顯示根據第 2B 圖內的 DTD 208，對應地從第 2A 圖之未結構化文件 200 所轉換的結構化文件 220。即如該圖所示，未結構化文件內的資料序列會被剖析成相關於定義在該結構化文件之 DTD 裡各文件單元的諸多資料單元。

不同於未結構化文件，經結構化文件可簡易地透過各文件單元而接取到某些資訊。結構化文件的表現方式通常是定義在個別的樣式文稿內，即如按曝序樣式文稿(CSS)，或是格式化物件之可延展樣式語言(XSL-FO)所撰



## 五、發明說明( )

寫者，這可解譯出各個文件單元的排置方式。這項功能可讓結構化文件透過不同樣式文稿，按不同媒體之不同排置方式而呈現。一般說來，該裝飾資訊或格式屬性，即如一未結構化文件裡的字型資訊，除非在 DTD 內被定義為文件單元屬性，否則會在將未結構化文件轉換成相對應的結構化文件之後即予棄除。進一步的格式資訊修飾通常不會影響到經轉換之結構化文件。

第 3A 圖說明根據本發明一具體實施例的功能性圖式 300。一轉換模組 302 係由一關聯模組 302 及一整合模組 306 所組成。該關聯模組 302 接收一未結構化文件，且最好是按超文件格式。與此同時，該關聯模組 302 也會接收一含有既已預先定義 DTD 的檔案，茲稱之為定義檔案。通常，DTD 係根據該未結構化文件的性質或目的所定義。例如，該未結構化文件係屬一接收範疇，即如第 2A 圖中的文件者，而在如第 2B 圖所示之定義檔案裡的 DTD 則是根據該“接收型態”文件所設計。

為進一步瞭解該關聯模組 302，第 3B 圖顯示一根據本發明之一具體實施例，實作該轉換模組 302 的環境 320。該

環境 320 包括兩個顯示畫面 322 與 324，以供使用者執行未結構化文件到按照標示語言之檔案(茲稱之為標示語言檔案)的轉換作業。該畫面 322 係用來顯示該未結構化文件。在一較佳具體實施例中，會載入該未結構化文件的超文件版本以供顯示。一起檔案，茲稱為未結構化文件或其

## 五、發明說明( )

列印版本，一般會含有許多可顯示物件。各個物件係字元、文字或圖形表現的簇集或群組。即如畫面 322 所示，各個文字或單項數碼係屬可顯示物件，而這會內定地被載送於該超檔案內。換句話說，各個物件是由許多包括像是，但不限於，該物件的型態、大小、色彩與位置等屬性或裝飾資訊所定義，使得該者可被“正確地”列印無誤。可由使用者按照彼等意義或目的，以人工方式將數個物件加以群組。例如該群組物件 326 包括三個字元型態物件“Green”、“Chili”及“Salsa”。自然地，這三個字元型態的物件可構成一個作為該群組物件 326 的標題。可對該畫面 322 內顯示之超檔案的其餘部分進行物件的群組作業。

畫面 324 是被用來顯示為該畫面 322 內顯示之超檔案所備製的定義檔案。為利於相關模組 302 的作業，該定義檔案係以圖形方式呈現為“DTD Pool” 328。例如，第 2B 圖內 DTD 208 的圖形表現 328 可被該畫面 324 用來說明各文件單元之間的層級關係。

根據一最終會將該超檔案轉換為 XML 檔案的具體實施例，會從該“DTD Pool” 328 產生出一輔助性 XML 樹狀物 330。該輔助性 XML 樹狀物 330 也會顯示出各文件單元之間的層級關係。此外，文件單元各者會被指配一包含，但不限於，如數字、名稱、字型、型態名稱或色彩的識別子。在一具體實施例中，該識別子係在文件單元各者的“資料”內。為將該群組物件 326 關聯於一文件單元

## 五、發明說明( )

“標題” 332，會在選定該群組物件 326 時即引動該“資料” 334。本發明之一特點，即在於可將該群組物件 326 相關到該“資料” 334 內之識別子的一種底含關聯性。特別是在一具體實施例中，如該“資料” 334 內的識別子為色彩者，即“綠色(green)”，則該群組物件 326 會被按綠色而強調，以表示出這個群組物件既已被關聯於該 DTD。假使識別子係一字型，“Arial”，則該群組物件 326 會被按 Arial 樣式而強調，以表示出這個群組物件既已被關聯於該 DTD。

類似地，群組物件 340 可按於“成分(ingredient)”，而關聯到該資料 342 內的識別子，群組物件 344 可按於“數量(amount)”，而關聯到該資料 346 內的識別子等等。因此，即已切割該畫面 322 內的超檔案，並可藉由一識別子將其內的可顯示物件各者關聯到所載入之 DTD 內的文件單元。該畫面 322 現具有一經修改之超檔案 310，而該者之一範例可如後文所詳述。

現再參考第 3A 圖，一經修飾之超檔案 310 被輸入到一整合模組 306 內，而又進一步接收一樣式文稿。樣式文稿通常會被組態設定以包括各項映對規則，這些映對規則是指根據顯現出從該超檔案而來的各個物件之媒體而定。一種示範性媒體為瀏覽器(如 Microsoft 公司所出售的 Internet Explorer)可接取之檔案的網頁表現方式。因之，該檔案係按如 HTML 或 XML 的標記語言所撰，茲稱為一標記語言檔案。

## 五、發明說明( )

第 3C 圖顯示一個這種依據該超檔案內的可顯示物件，按 XML 格式所設計之樣式文稿的範例。一般說來，一樣式文稿係經設計以個別地定出各者物件的位置、色彩或大小，俾對某特定媒體達到適當且引人注目的表現方式。第 3C 圖中的範例係經設計以呈現“食譜型態”文件，並當載入時可讓該經修飾的超檔案產生一個適當的 XML 檔案。換句話說，該整合模組 306 可根據該樣式文稿，而從該經修飾的超檔案產生出 XML 檔案。按在此所給定的說明，應即注意到並不需要將該樣式文稿輸入到該整合模組 306 內。在一種實作方式裡，可將該等映對規則載入於該 DTD 檔案，使得該整合模組 306 根據所載入的 DTD 檔案，來執行從該經修飾的超檔案到一標記語言檔案的映對作業。

根據一具體實施例，該轉換模組 302 係按軟體所實作，並可按如一應用程式而配發給各使用者或服務供應商。應明瞭這項從一未結構化文件到一標記語言檔案的轉換處理實不易按照成本斷定方式予以量訂化。在該轉換模組 302 中含有一計數器 308。在一具體實施例裡，該計數器 308 係經設定以計算該超檔案內待加轉換的網頁數。只要每次當在畫面（如頁面畫面）內所有的物件皆被關聯到一 DTD 檔案中的各文件單元，並按為相對應的經修飾之超檔案所存檔時，該計數器 308 就會遞增數值。第 3D 圖顯示一保存於一 dongle 內的計數結果範例。Dongle（發音為 DONG-uhl）是一種機制，用以確保僅經授權之使用者可

## 五、發明說明( )

拷貝或使用一特定軟體應用程式，尤其是極為昂貴的軟體。常見的 dongle 實作方式包括插置於電腦的平行或序列機埠上之硬體鑰匙，而該軟體會在繼續執行之前先對其接取辨識；亦按類似方式所接取的特殊碟片鑰匙；以及在工廠或於系統設定過程中被載入某種形式之唯讀記憶體內的註冊編號。

當需要重置該 dongle 時，可按成本算定的方式來評估該轉換處理。根據一具體實施例，含有該轉換模組 302 實作之產品的所有主，可按免費或極低成本的方式將該產品配發給各使用者。通常，使用者會需要從各種編修工具所組創、編輯或管理的未結構化文件中產生許多網頁目錄。使用者按此方式來接收產品之一優點在於，在使用該產品前並不需要籌備大筆的資金以取得該產品。各使用者可僅支付使用該產品的費用。從而，利用具有轉換模組 302 之 dongle 其一目的就在於管理其使用權。因此，該產品的所有主即可藉由控制含有使用資訊的 dongle，來掌握該產品的使用權。

第 3E 圖為一利用含有根據本發明一具體實施例之轉換模組 302 實作產品的方法流程 370 圖式。有時候，是由使用者或一企業體按租賃方式來取用產品。或者，產品是由提供服務項目給需要將未結構化文件轉換成按於不同媒體表現(即如呈現在一網站上)結構化文件的各個企業之服務供應商所使用。

處理 370 開始於 372 處，即從經編修文件產生超檔

## 五、發明說明( )

案。通常，或利用一種或更多的編修工具來備置該經編修文件。即如前述，最好是從該經編修文件取得該等超檔案，這可使得該轉換模組 302 不需要針對各個不同的編修工具來個別地組態設定。但是，應注意傾向於利用超檔案並不屬於本發明的內定限制，而是可令該轉換模組 302 在運作上更具有效益性。對於熟諳本項技藝之人士而言，應可瞭解可組態設定一轉換介面或一系列印驅動器，以容納任何款式的經編修文件或產生該等超檔案。

一旦取得該等超檔案之後，即可將彼等載入一視像環境內，其中可個別地顯示這些超檔案。第 3B 圖的環境 320 即屬適用者，在此可個別載入這些超檔案各個頁面以供顯示。

在該處理 370 允許使用者進行下一步之前，會先引動一授權處理 378 以確保該使用者操作的是既經授權之產品。即如前述，一種示範性授權方法係透過提供/擁有/掌控該產品之企業或經銷商所預先設定的 dongle 而進行。如果該授權處理 378 指示出該處理 370 非經授權者，則通常會向使用者顯示出一個畫面以說明需如何授權該產品。設定該授權 376 的各程序之一牽涉到所購買的轉換或儲存允許頁數。

根據一具體實施例，可利用一 dongle 以耦接至一執行該處理 370 的電腦處。該 dongle 含有第一及第二數值。該第一數值係一起始數，如“10”，而第二數值係一限制數，如“1000”，這表示可由該處理 370 處理及儲存 1000

## 五、發明說明( )

頁的轉換文件。

一旦該處理 370 既獲允許前進到 380，該使用者現可根據例如其意義或目的並按照所載入的 DTD 檔案，分別地集設數個可顯示物件而為多個群組物件，並顯示於鄰近處。在 382 處，該等群組物件可分別地被關聯到該 DTD 檔案內的各項定義。至少部分的定義會具有數個識別子，且最好是各個識別子會被關聯或指配給各識別子其中之一者為宜。

即如前釋，會將該既選物件與定義間的各關聯性收存於一經修飾的超檔案內。而在該處理 370 允許進行該項儲存之前，會於 386 處先檢查一計數器。在一具體實施例中，會比較 dongle 內的第一及第二數值。當該第一數值大致接近該第二數值時，例如兩個相同數值，，則該處理 370 會詢問以補充該允許用量。通常，使用者需令由依照該 dongle 內之資訊而收費的企業或經銷商以重置或重新設定該 dongle。在 384 處，數值現經重置並可允許該處理 370 前進。

在 386 處，可進行一儲存步驟。依據真正的實作方式而定，可將經修飾的超檔案或一標記語言檔案收存在一儲存空間內。該標記語言檔案是按照一預定媒體表現方式，參酌於樣式文稿而自該經修飾超檔案所產生。

應注意到在第 3E 圖中，是在 386 處，特別是在 384 之後，才檢查該計數器。事實上，對於熟諳本項技藝之人士而言，可明顯知悉確得沿該處理 370 內幾乎任處來檢查

## 五、發明說明( )

或詢知該計數器。在此利用一計數器的目的，係為助於業者控制並斷定該處理 370 的使用狀況，藉此決算出成本而依之收費。

第 4 圖顯示一根據本發明一具體實施例之資料處理模組 404 的功能性區塊圖式 400。包含在該整合模組 306 內的資料處理模組 404 含有一輸入模組 406、一編輯模組 410 及一轉換或過濾模組 414。該資料處理模組 404 所執行的各項功能其中之一係為將未結構化文件或具不同 DTD 的結構化文件，轉換成具預定或特定 DTD 之相對應結構化文件。

該輸入模組 406 從文件資料庫 402 處載入文件或輸入文件，該文件資料庫即對應於如第 1A 圖內計算裝置 102 裡的存庫。或另者，該輸入模組 406 可開啟新的文件 408。應注意到所載入或輸入的文件可以是未結構化(如一超檔案)或是結構化文件，並且在某些情況下可具有一些預創的結構性字型資訊。

該編輯模組 410 可與該輸入模組 406 相通訊，並產生/編輯該等輸入文件的結構性字型資訊。這個模組可供選取輸入文件的資料單元，並提供一編輯環境以替換所選定之各資料單元的字型屬性，像是字型款式、字型樣式、字型色彩、字型大小和字型效果等。將該等輸入文件剖析成諸資料單元並且指配各字型屬性的方式，係依照如所欲 DTD 及相關字型屬性 412 內所定義的各文件單元之相關性表單而定。第 5 圖中繪示對於該 DTD 412 的示範性相關性表單



## 五、發明說明( )

500，其中含有文件單元 502、元件屬性 504、字型款式 506、字型樣式 508、字型色彩 510、字型大小 512 和字型效果 514 等欄位。

第 6 圖顯示如第 2A 圖之未結構化文件 200 的編輯結果 600。經剖析的資料單元或經合併的物件 602、604、606、608、610、612 與 614 各者既已根據如第 5 圖的相關性表單元而指配字型屬性，並分別地按照相關的字型而顯示。在剖析過程中，本模組可供允依據該輸入文件 602 的讀取順序來進行資料單元的序列選取，俾編輯其等的字型資訊。這個模組也可供資料單元的區域群組以編輯其等的字型資訊。這個模組亦可提供該相關性表單的輔助性審視方式。

該轉換或過濾模組 414 可利用該結構性字型資訊，將所載入的文件轉換成具使用者定義之文件型態定義(DTD)的結構化文件。在此，會將依據該等文件單元及轉換 416 之字型屬性的各項映對規則輸入或設計於本模組內。

第 7 圖說明該等用以將經編輯文件 602 轉換成如第 2C 圖結構化文件 220 之映對規則的範例 700。尤其是，702 開始且 718 結束該“文件”單元，704 開始且 714 結束該“食譜”文件，706 構成該“成分”單元，708 構成該“程序”單元，710 構成該“表現”單元而 712 構成該“源自”單元。在 704、706、708、710 和 712 中，該結構性字型資訊會被用來定位該等資料單元，而經定位的資料單元會被指配為各文件單元的屬性或經剖析之字元資料。可

## 五、發明說明( )

藉如，但不限於，Java、JavaScript、可延展性轉換作業樣式語言(XSLT)、C/C++等程式語言，或是任何內建或程式化硬體裝置來實作這些映對規則。經轉換的文件可儲存為一檔案文件，或是輸出到該文件資料庫418處。

除直接轉換成所欲之文件外，該轉換模組也可將經編輯文件輸出為一中介結構化檔案，而這會含有結構性的字型資訊。可重新載入該中介結構化檔案，以供進一步的編輯或批次轉換作業。

該經編輯文件600的中介結構化檔案808之一範例可如第8圖所示，其中802含有具特定字型屬性之“字型”單元，而804含有具“font\_ID”屬性的經剖析資料單元，藉以關聯到該字型資訊。由於具相同字型屬性的經剖析資料單元既已被該中介結構化檔案裡相同的“font\_ID”所集組，故可根據經分群的字型資訊來設計該等轉換作業映對規則。

第9圖說明一用以將該中介文件800轉換成該結構化文件220的映對規則範例。特別是，902開始且918結束該“文件”單元，904開始且914結束該“食譜”文件，906構成該“成分”單元，908構成該“程序”單元，910構成該“表現”單元而912構成該“源自”單元。在904、906、908、910和912中，經集組的字型資訊會被用來定位該等資料單元，而經定位的資料單元會被指配為各文件單元的屬性或經剖析之字元資料。

第10圖顯示利用可延展性轉換作業樣式語言

## 五、發明說明( )

(XSLT)，實作出該等如第 9 圖所給定之映對規則的範例。

前述之發明最好以按軟體、硬體或其等組合實作為佳。本發明至少部分可實作為電腦可讀取媒體上的電腦可讀取程式碼。該電腦可讀取媒體是任何可存放資料而於後由計算裝置所讀出的資料儲存裝置。該等電腦可讀取媒體的範例包括唯讀記憶體、隨機存取記憶體、硬碟機、軟碟、光碟機(CD-ROM)、DVD、磁帶、光學資料儲存裝置、載波。該電腦可讀取媒體亦可散佈於網路耦接的電腦系統上，而按散佈方式來儲存及執行該電腦可讀取程式碼。

本發明既已按足夠細節而具相當程度特定性所陳述。然對於熟諳本項技藝之人士而言，應可瞭解本具體實施例揭示僅屬範例性質，且得執作各部分的排置及組合方面之各種變化，而無虞悖離如後申宣之本發明精神及範圍。在此所論述的各項具體實施例雖似含有對於資訊單位表現方式的某些限制，然就以格式及排置方式而言，本發明確亦可良好適用於該等具體實施例以外者，熟諳本項技藝之人士對此應為顯易知悉。因之，本發明範圍應按後載之申請專利範圍，而非前揭各具體實施例說明所定義。

(請先閱讀背面之注意事項再填寫本頁)

訂

線

## 六、申請專利範圍

- 1.一種用以產生結構化文件的方法，該方法至少包含下列步驟：

接收一含有文件型態定義(DTD)的定義檔案；

並將一超檔案且連同該定義檔案予以顯示，其中該超檔案含有數個可顯示物件，及其關於該等可顯示物件各者的個別裝飾屬性；及

將該定義檔案內各定義至少其一關聯於可顯示物件之一者。

- 2.如申請專利範圍第1項所述之方法，其中更包含下列步驟：

產生一經修飾之超檔案，該者包括該等可顯示物件，各個被關聯於該定義檔案內各定義至少一者。

- 3.如申請專利範圍第2項所述之方法，其中更包含根據一組映對規則而將該經修飾之超檔案轉換為一標記語言檔案的步驟。

- 4.如申請專利範圍第1項所述之方法，其中上述之定義檔案包括一文件單元結構，各者對應於該超檔案內各可顯示物件之一者。

- 5.如申請專利範圍第4項所述之方法，其中部分的文件單元包含另一層級的子文件單元，各個子文件單元對應於

## 六、申請專利範圍

該超檔案內各可顯示物件之一者。

6.如申請專利範圍第4項所述之方法，其中至少部分的文件單元分別地包含數個識別子，各個識別子係被指配給至少部分的文件單元其中一者。

7.如申請專利範圍第6項所述之方法，其中部分的識別子係其一或更多的數碼或字母。

8.如申請專利範圍第6項所述之方法，其中部分的識別子係從一含有一字型款式、一色彩名稱、一大小、一樣式及一效果之群組中所選出。

9.如申請專利範圍第6項所述之方法，其中相關連該定義檔案內的定義至少一者的步驟至少包括下列步驟：  
選取可顯示物件之一者；及  
將其一識別子指配給該選定可顯示物件。

10.如申請專利範圍第9項所述之方法，其中該等識別子其一係一數碼或字母。

11.如申請專利範圍第10項所述之方法，其中該等識別子其一係一個或更多的(i)字型款式，(ii)色彩名稱，(iii)大小，(iv)樣式及(v)效果。

## 六、申請專利範圍

12.如申請專利範圍第 1 項所述之方法，其中該超檔案係屬，或產源自，由一編修工具所組創、編輯或管理的未結構化文件。

13.如申請專利範圍第 12 項所述之方法，其中部分的可顯示物件係個別的字元群組。

14.如申請專利範圍第 13 項所述之方法，其中部分的裝飾屬性至少包括該字元群組裡各者的位置、字型色彩、字型大小、字型款式、樣式及效果。

15.一種用以產生結構化文件的方法，該方法至少包含下列步驟：

啟動一含有第一畫面及第二畫面的環境，該第一畫面顯示一超檔案而該第二畫面顯示一含有文件型態定義(DTD)的定義檔案，其中該超檔案含有數個可顯示物件，及其關於該等可顯示物件各者的個別裝飾屬性，且其中文件型態定義各個包含一識別子；

群集出數個群組物件，各個群組物件含有數個可顯示物件；及

將各個群組物件關聯於該等文件型態定義其一內的識別子。

16.如申請專利範圍第 15 項所述之方法，其中更包含產生

## 六、申請專利範圍

一經修飾之超檔案，該超檔案含有群組物件各個的資訊，而該者係被關聯於各文件型態定義其一內之識別子。

17.如申請專利範圍第 16 項所述之方法，其中更包含下列步驟：

根據映對規則，將該經修飾之超檔案轉換為一標記語言檔案。

18.如申請專利範圍第 17 項所述之方法，其中該標記語言檔案適合表現於一選定媒體上。

19.如申請專利範圍第 18 項所述之方法，其中該選定媒體係網際網路上的網頁表現。

20.如申請專利範圍第 18 項所述之方法，其中該標記語言檔案係根據由一群組所選出的標記語言，該群組包含“超文字標記語言(HTML)”、“精簡超文字標記語言(cHTML)”、“可延展標記語言(XML)”、“標準通用標記語言(SGML)”或是“無線標記語言(WML)”。

21.如申請專利範圍第 15 項所述之方法，其中部分的裝飾屬性至少包括該字元群組裡各者的位置、字型色彩、字型大小、字型款式、樣式及效果。

## 六、申請專利範圍

22.如申請專利範圍第 21 項所述之方法，其中部分的可顯示物件係個別的字元群組。

23.如申請專利範圍第 22 項所述之方法，其中部分的識別子係其一或更多的數碼和字母。

24.如申請專利範圍第 23 項所述之方法，其中該等識別子其一係一個或更多的(i)字型款式，(ii)色彩名稱，(iii)大小，(iv)樣式及(v)效果。

25.一種機器可讀取媒體，實作供一處理器執行之各項指令，當該處理器執行時，這些指令會啟引該處理器產生結構化文件，該機器可讀取媒體至少包含：

一第一程式碼，用以接收一含有文件型態定義(DTD)之定義檔案；

一第二程式碼，用以顯示一超檔案併連同該定義檔案，該超檔案含有數個可顯示物件，及其關於該等可顯示物件各者的個別裝飾屬性；及

一第三程式碼，用以將該定義檔案內至少一定義關聯於該等可顯示物件之一。

26.如申請專利範圍第 25 項所述之機器可讀取媒體，其中更包含：



## 六、申請專利範圍

一 第四程式碼，用以產生一經修飾之超檔案，該超檔案含有該等可顯示物件，而各者係被關聯於該定義檔案內至少一定義。

27.如申請專利範圍第 25 項所述之機器可讀取媒體，其中更包含第五程式碼，用以根據映對規則，將該經修飾之超檔案轉換為一標記語言檔案。

28.如申請專利範圍第 25 項所述之機器可讀取媒體，其中上述之定義檔案包括一文件單元結構，各者對應於該超檔案內各可顯示物件之一者。

29.如申請專利範圍第 28 項所述之機器可讀取媒體，其中部分的文件單元包含另一層級的子文件單元，各個子文件單元對應於該超檔案內各可顯示物件之一者。

30.如申請專利範圍第 28 項所述之機器可讀取媒體，其中至少部分的文件單元分別地包含數個識別子，各個識別子係被指配給至少部分的文件單元其中一者。

31.如申請專利範圍第 30 項所述之機器可讀取媒體，其中部分的識別子係其一或更多的數碼或字母。

32.如申請專利範圍第 30 項所述之機器可讀取媒體，其中

## 六、申請專利範圍

部分的識別子係從一含有一字型款式、一色彩名稱、一大小、一樣式及一效果之群組中所選出。

33.如申請專利範圍第30項所述之機器可讀取媒體，其中該第三程式碼至少包括：

一第六程式碼，用以選取可顯示物件之一者；及

一第七程式碼，用以將其一識別子指配給該選定可顯示物件。

34.如申請專利範圍第33項所述之機器可讀取媒體，其中該等識別子其一係一數碼或字母。

35.如申請專利範圍第34項所述之機器可讀取媒體，其中該等識別子係(i)字型款式、(ii)色彩名稱、(iii)大小、(iv)樣式及(v)效果之一者或多者。

36.如申請專利範圍第25項所述之機器可讀取媒體，其中該超檔案係屬，或產源自，由一編修工具所組創、編輯或管理的未結構化文件。

37.如申請專利範圍第36項所述之機器可讀取媒體，其中部分的可顯示物件係個別的字元群組。

38.如申請專利範圍第37項所述之機器可讀取媒體，其中

## 六、申請專利範圍

部分的裝飾屬性至少包括該字元群組裡各者的位置、字型色彩、字型大小、字型款式、樣式及效果。

39.一種機器可讀取媒體，實作供一處理器執行之各項指令，當該處理器執行時，這些指令會啟引該處理器產生結構化文件，該機器可讀取媒體至少包含：

一第一程式碼，用以啟動一含有第一畫面及第二畫面的環境，該第一畫面顯示一超檔案而該第二畫面顯示一含有文件型態定義(DTD)的定義檔案，其中該超檔案含有數個可顯示物件，及其關於該等可顯示物件各者的個別裝飾屬性，且其中文件型態定義各個包含一識別子；

一第二程式碼，用以群集出數個群組物件，各個群組物件含有數個可顯示物件；及

一第三程式碼，用以將各個群組物件關聯於該等文件型態定義其一內的識別子。

40.如申請專利範圍第 39 項所述之機器可讀取媒體，其中更包含一第四程式碼，用以產生一經修飾之超檔案，該超檔案含有群組物件各個的資訊，而該者係被關聯於各文件型態定義其一內之識別子。

41.如申請專利範圍第 40 項所述之機器可讀取媒體，其中更包含一第五程式碼，用以根據映對規則，將該經修飾之超檔案轉換為一標記語言檔案。

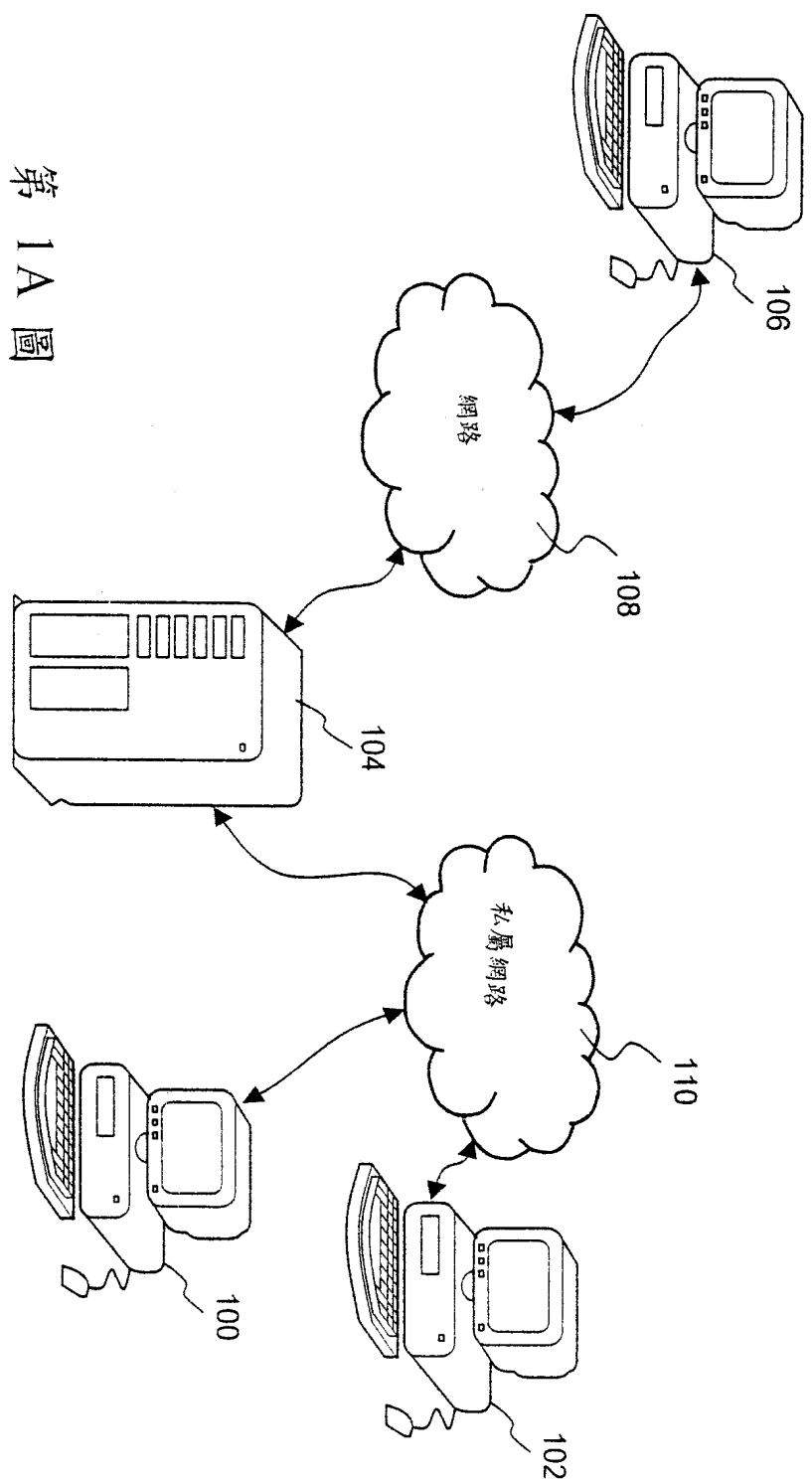
## 六、申請專利範圍

42.如申請專利範圍第 39 項所述之機器可讀取媒體，其中部分的裝飾屬性至少包括該字元群組裡各者的位置、字型色彩、字型大小、字型款式、樣式及效果，且其中部分的可顯示物件係個別的字元群組。

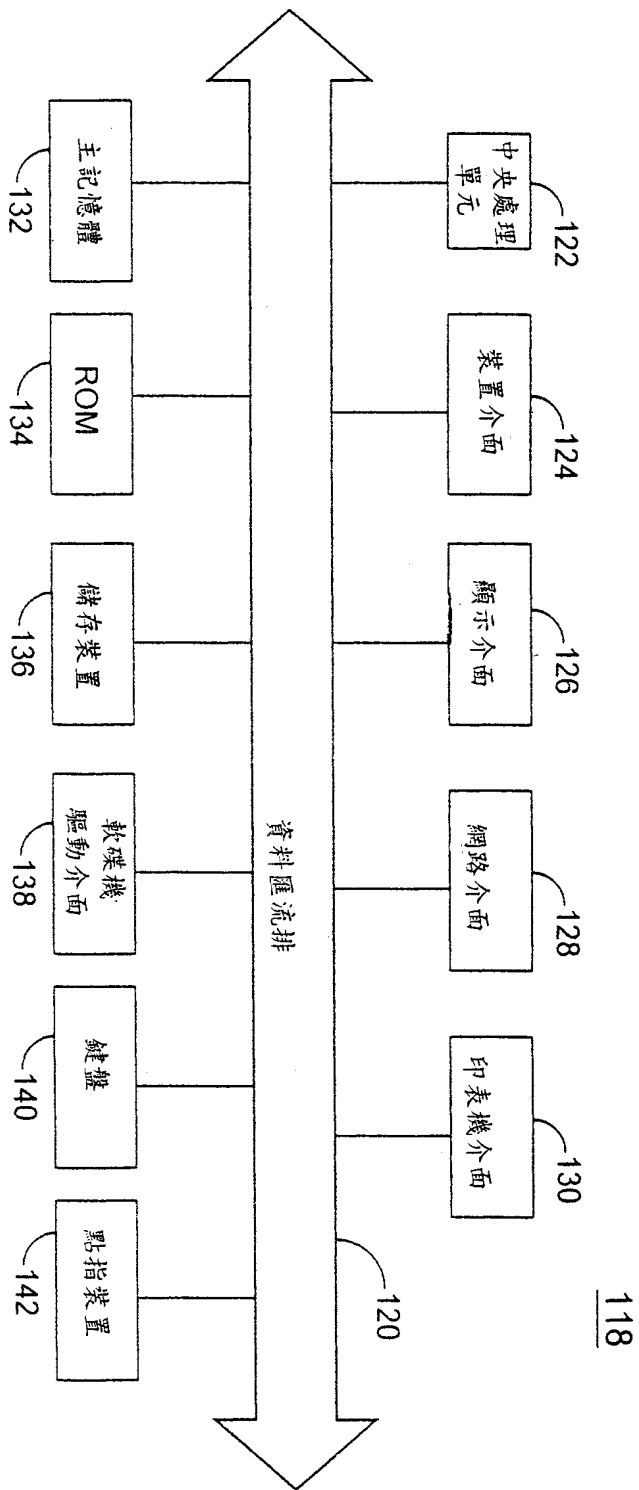
(請先閱讀背面之注意事項再填寫本頁)

訂

線



第 1A 圖



第 1B 圖

200

202

Green Chili Salsa

Chopped Tomatoes	1 16-ounce Can
Chopped Green Chile	3 Tablespoons
Diced Onions	3 Tablespoons
Chopped Garlic	1 Tablespoon
Salt	¼ Teaspoon
Lime Juice	½ Lime
Chopped Cilantro	1 Tablespoon

In a medium bowl, mix together all ingredients. Use as a dip, or over grilled vegetables, or toss with bowtie noodles for an appetizer course.

[www.VeggieLife.com](http://www.VeggieLife.com)

November 1997

```
<!--..... Document Type Definition (DTD) .....-->
<!-- document element -->
<ELEMENT document (recipe+)>
```

```
<!--..... recipe element .....-->
<ELEMENT recipe (ingredient+, procedure+, presentation, originate)>
<ATTLIST recipe title CDATA #REQUIRED>
```

```
<!--..... ingredient element .....-->
<ELEMENT ingredient (#PCDATA)>
<ATTLIST ingredient amount CDATA #REQUIRED>
```

```
<!--..... procedure element .....-->
<ELEMENT procedure (#PCDATA)>
<ATTLIST procedure step CDATA #IMPLIED>
```

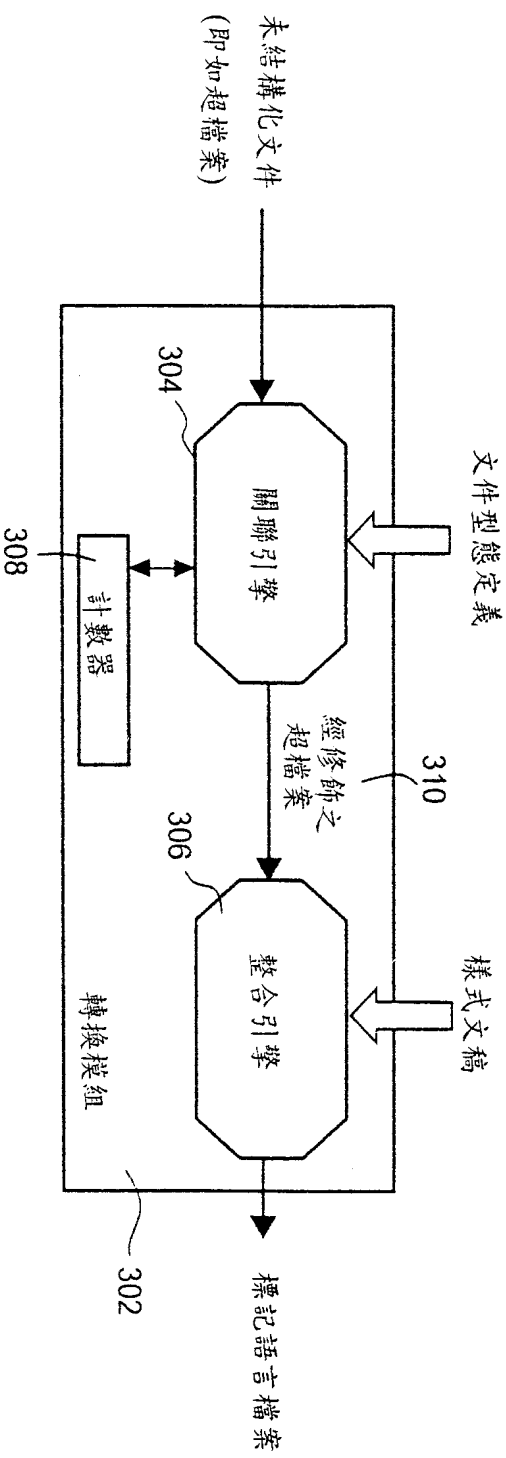
```
<!--..... presentation element .....-->
<ELEMENT presentation (#PCDATA)>
```

```
<!--..... originate element .....-->
<ELEMENT originate EMPTY>
<ATTLIST originate name CDATA #REQUIRED>
<ATTLIST originate website CDATA #IMPLIED>
<ATTLIST originate date CDATA #REQUIRED>
<!--..... END of DTD .....-->
```

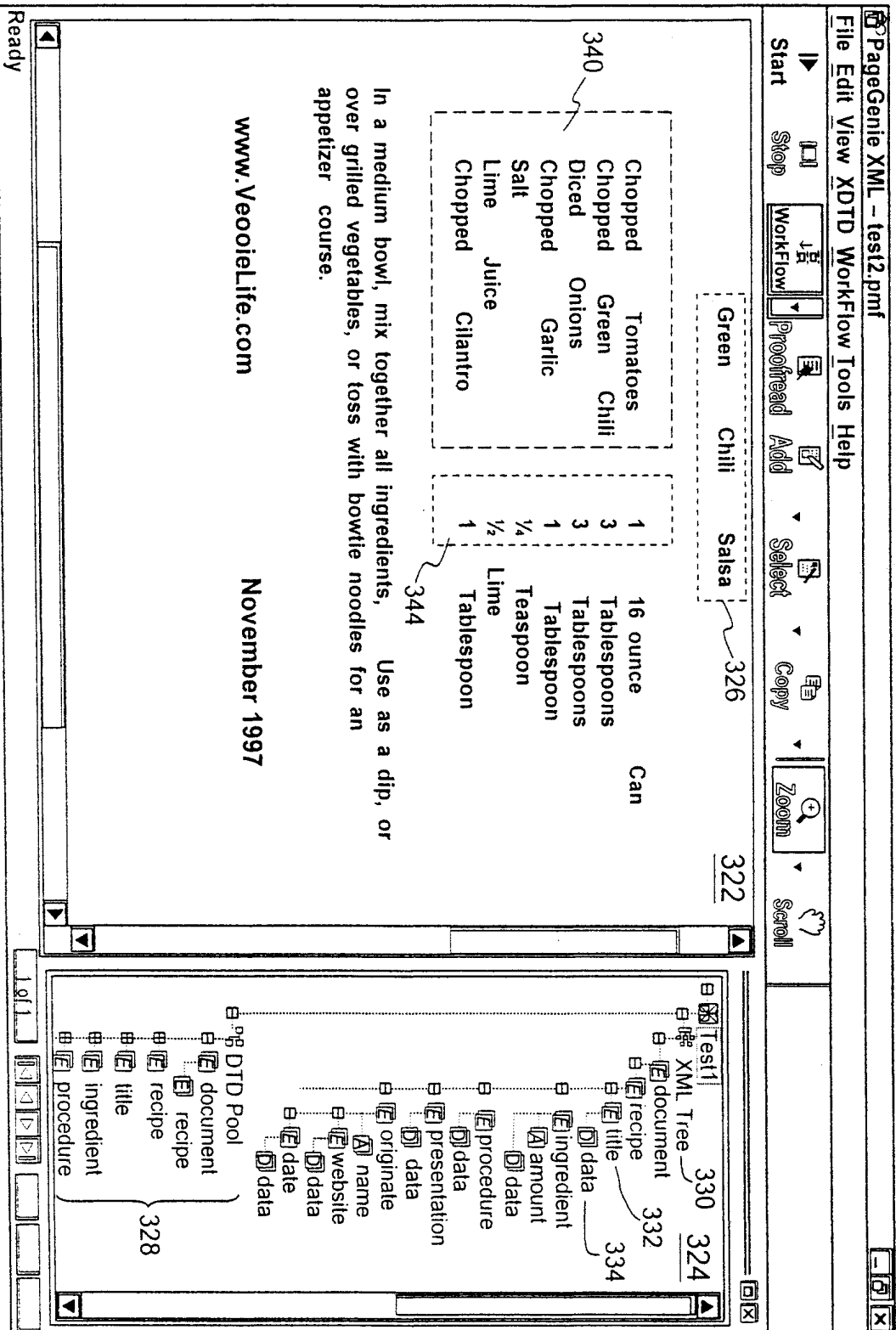
第 2B 圖



```
<document>
<recipe title="Green Chile Salsa">
  <ingredient amount="1 16-ounce Can">Chopped Tomatoes</ingredient>
  <ingredient amount="3 Tablespoons">Chopped Green Chile</ingredient>
  <ingredient amount="3 Tablespoons">Diced Onions</ingredient>
  <ingredient amount="1 Tablespoon">Chopped Garlic</ingredient>
  <ingredient amount="1/4 Teaspoon">Salt</ingredient>
  <ingredient amount="1/2 Lime">Lime Juice</ingredient>
  <ingredient amount="1 Tablespoon">Chopped Cilantro</ingredient>
  <procedure>In a medium bowl, mix together all ingredients.</procedure>
  <presentation>Use as dip, or over grilled vegetables, or toss with bowtie noodles for an appetizer course.</presentation>
  <originate name="VeggieLife" website="www.VeggieLife.com" date="November 1997" />
</recipe>
</document>
```



第 3A 圖



## 第 3C 圖

```

<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/" >
    <html>
      <body>
        <xsl:apply-templates />
      </body>
    </html>
  </xsl:template>

  <xsl:template match="document">
    <xsl:apply-templates />
  </xsl:template>

  <xsl:template match="recipe">
    <h2><center><font color="teal">Title: </font><font color="red">
      <xsl:value-of select="title"/>
    </font></center></h2>
    <h3><center><font color="teal">From: </font><font color="gray">
      <xsl:for-each select="originate" >
        <xsl:value-of select="@name" />
      </xsl:for-each>
    </font></center></h3>
    <p><center><table bgcolor="ffffc0" cellspacing="12"><tr><td>
      <xsl:for-each select="image">
        <IMG SRC="recipe.jpg" HEIGHT="200" WIDTH="220"/></xsl:for-each></td><td><table>
          <tr><td><font color="teal" size="5">ingredient:</font></td></tr>
          <xsl:for-each select="ingredient">
            <tr><td><xsl:value-of select="text()" /></td>
            <td><font color="blue"><xsl:value-of select="amount"/></font></td></tr>
          </xsl:for-each>
        </table></td></tr></table></center></p>

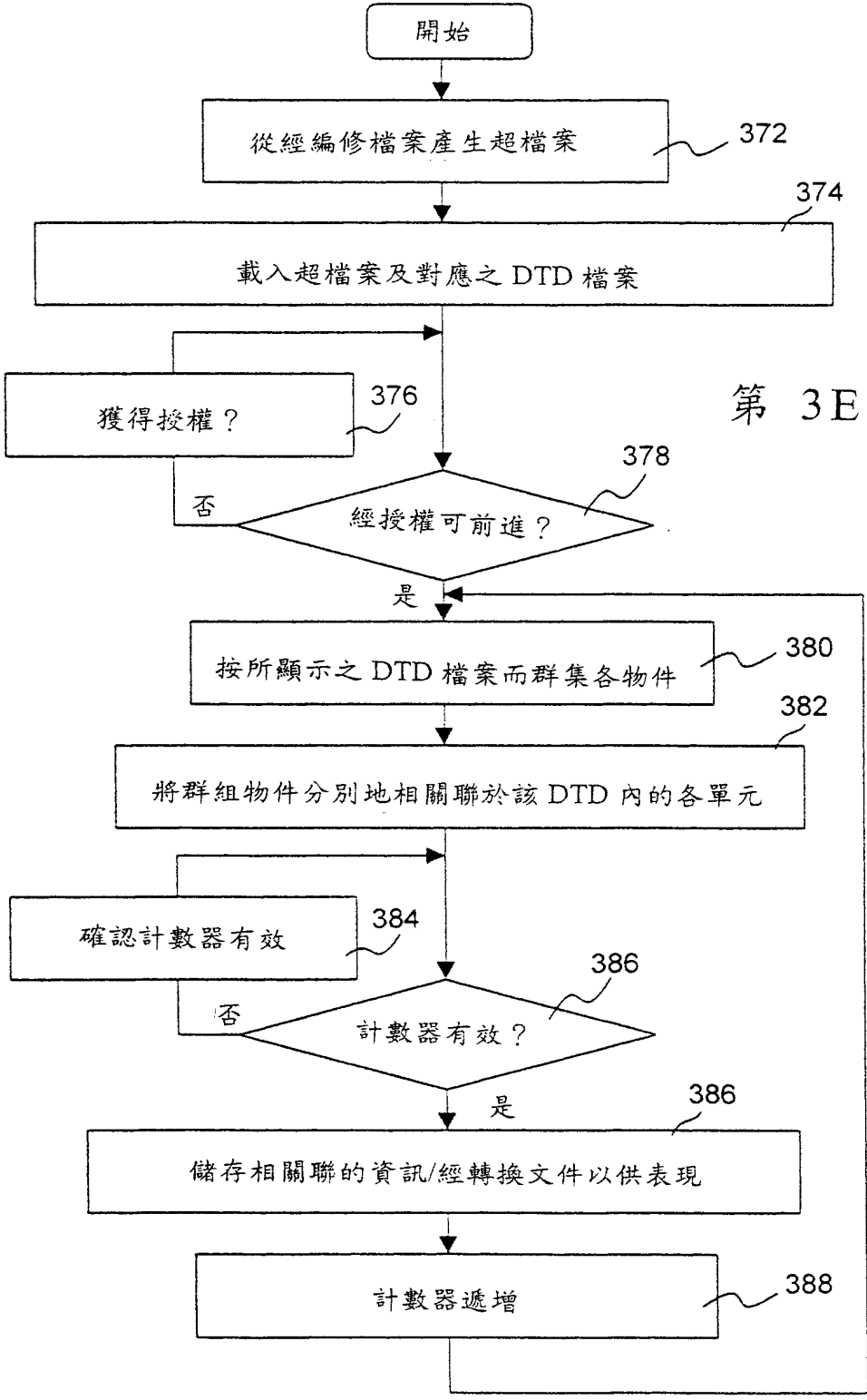
    <p><center><table><tr><td><font color="teal" size="5">Procedure:</font></td></tr>
      <xsl:for-each select="procedure">
        <tr><td><xsl:value-of select="text()" /> </td></tr>
      </xsl:for-each>
      <tr><td></td></tr>
      <tr><td><font color="teal" size="5">Presentation:</font></td></tr>
      <xsl:for-each select="presentation">
        <tr><td><xsl:value-of select="text()" /> </td></tr>
      </xsl:for-each>
    </table></center></p>

    <p><center><table bgcolor="c0ffff" width="80%">
      <xsl:for-each select="originate">
        <tr><td align="left"><xsl:value-of select="website" /></td>
        <td align="right"><xsl:value-of select="date"/></td></tr>
      </xsl:for-each>
    </table></center></p>
  </xsl:template>
</xsl:stylesheet>

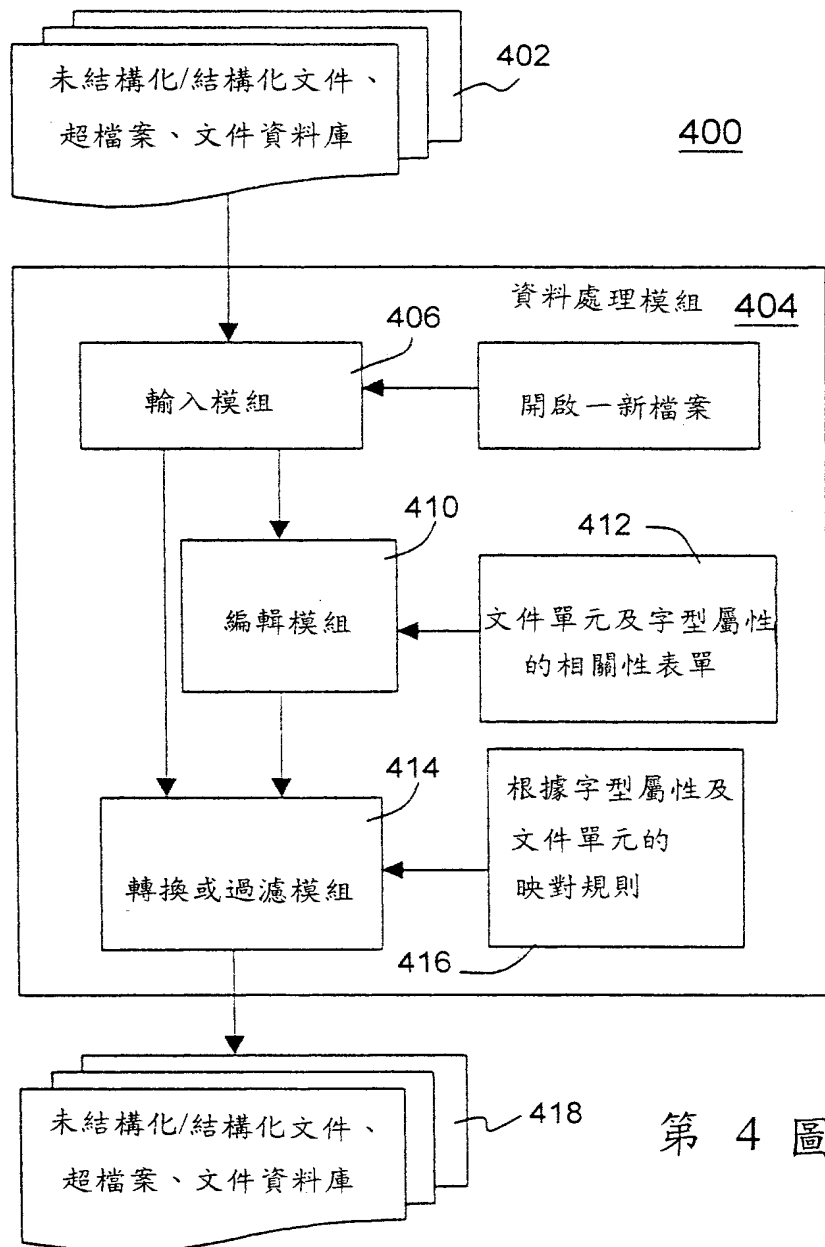
```

PageGenie XML 的 Dongle 資訊			
已存檔數：71		限數：1000	
編籤計數：51		限數：1000	
Dongle, 建立於西元 2000 年 8 月 12 日			

第 3D 圖



第 3E 圖

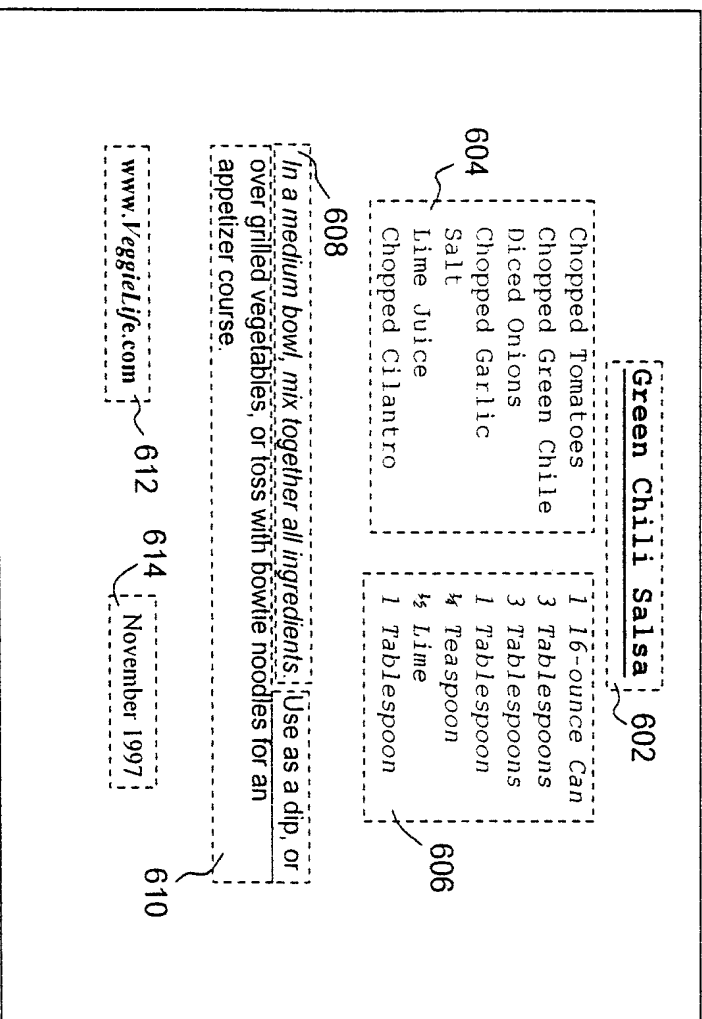


第 4 圖

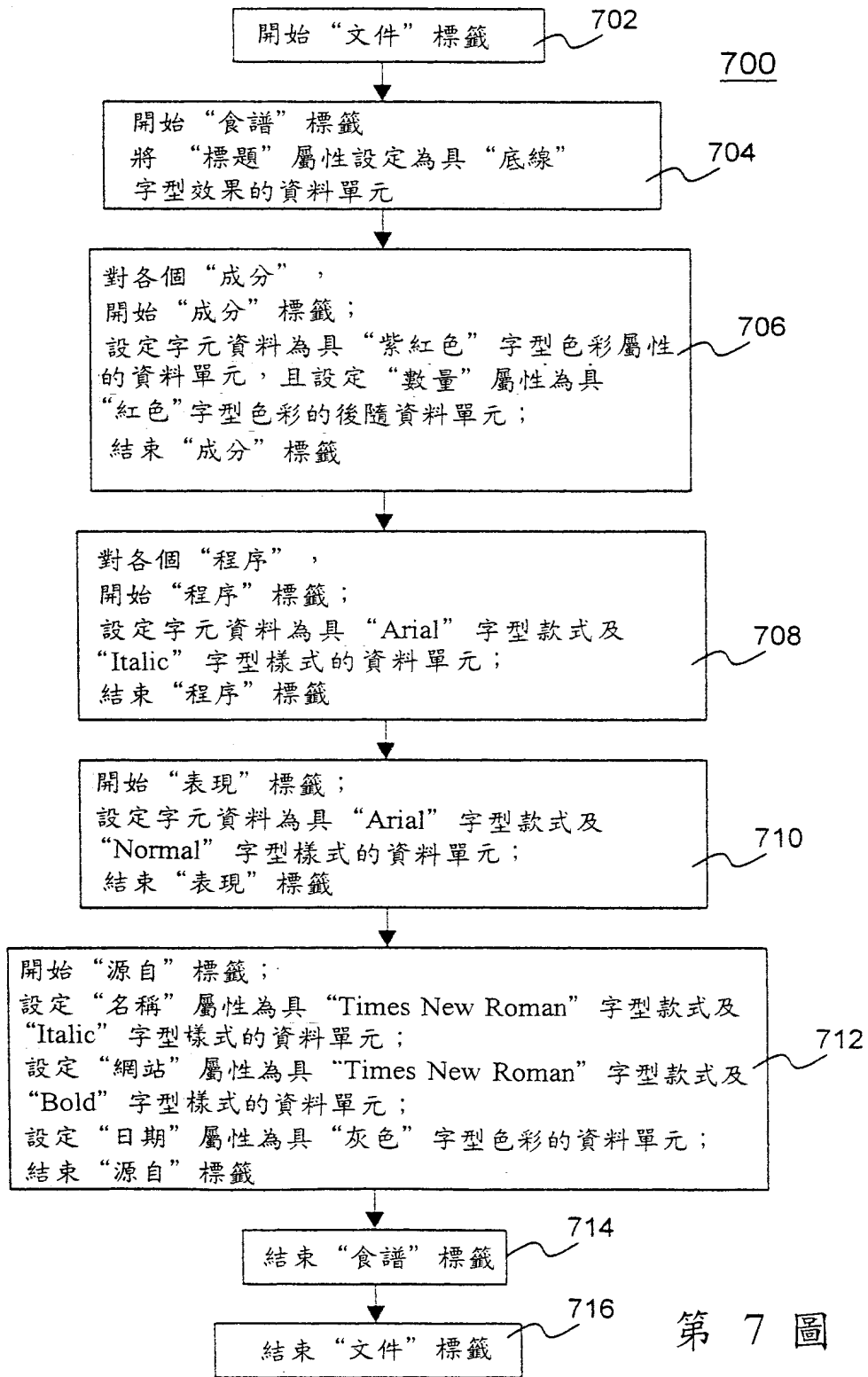
502		504		506		508		510		512		514	
文件單元	元件屬性	字型款式	字型樣式	字型色彩	字型大小	字型效果							
recipe	title	Courier New	Bold	Black (0x000000)	12	Underline							
ingredient		Courier New	Normal	Magenta (0xFF00FF)	10	Normal							
ingredient	amount	Courier New	Italic	Red (0x0000FF)	10	Normal							
procedure		Arial	Italic	Blue (0xFF0000)	10	Normal							
presentation		Arial	Normal	Cyan (0xFFFF00)	10	Normal							
originate	name	Times New Roman	Bold	Green (0x00FF00)	10	Normal							
originate	website	Times New Roman	Bold + Italic	Yellow (0x00FFFF)	10	Normal							
originate	date	Times New Roman	Normal	Gray (0x808080)	10	Normal							

第 5 圖





第 6 圖



第 7 圖

```
<document>
<font font_ID="1" type="Courier New" style="Bold" color="0x000000" size="12" effect="Underline" />
<font font_ID="2" type="Courier New" style="Normal" color="0xFF00FF" size="10" effect="Normal" />
<font font_ID="3" type="Courier New" style="Italic" color="0x0000FF" size="10" effect="Normal" />
<font font_ID="4" type="Arial" style="Italic" color="0xFF0000" size="10" effect="Normal" />
<font font_ID="5" type="Arial" style="Normal" color="0xFFFF00" size="10" effect="Normal" />
<font font_ID="6" type="Times New Roman" style="Bold" color="0x00FF00" size="10" effect="Normal" />
<font font_ID="7" type="Times New Roman" style="Bold Italic" color="0x00FFFF" size="10" effect="Normal" />
<font font_ID="8" type="Times New Roman" style="Normal" color="0x808080" size="10" effect="Normal" />

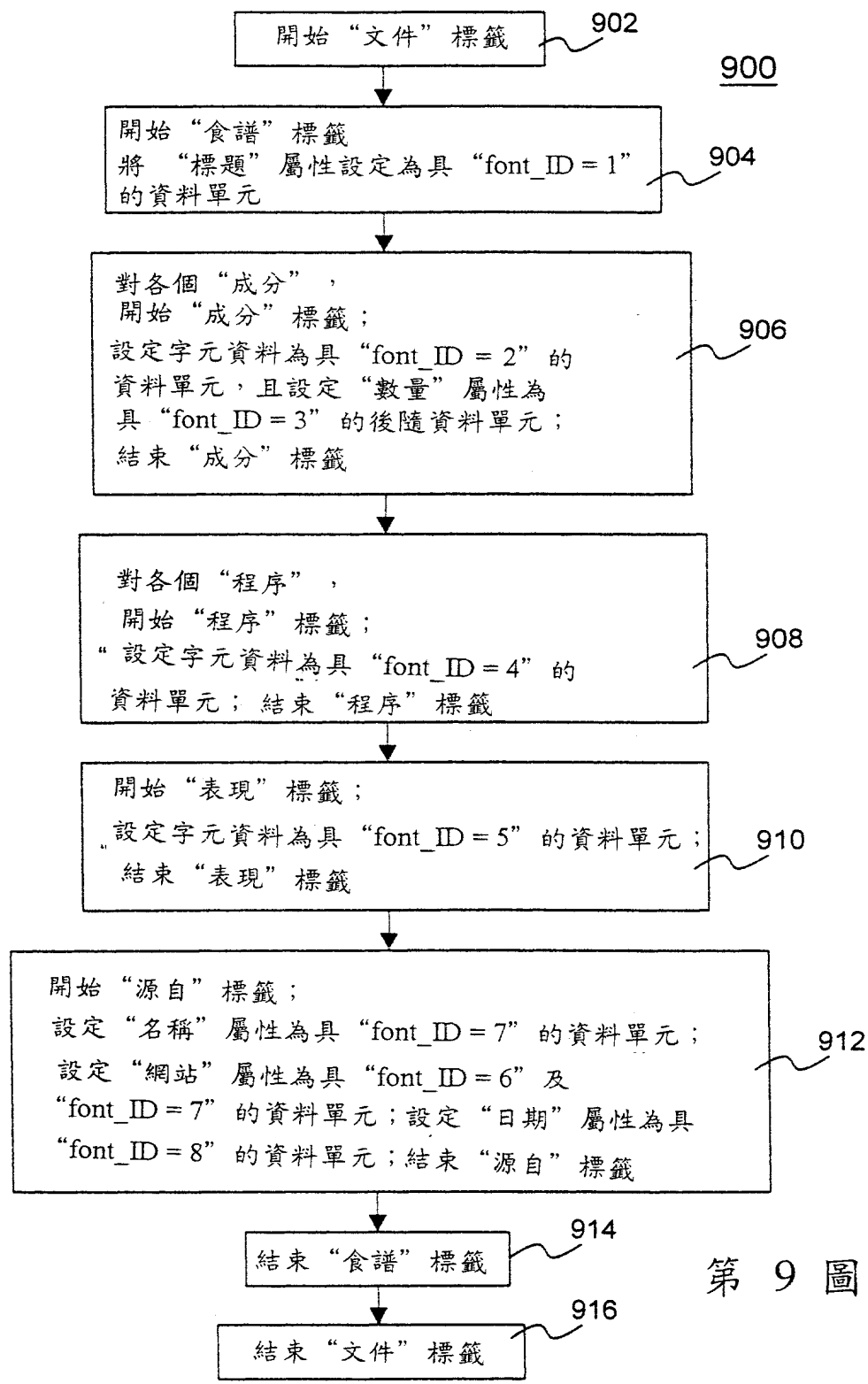
<data font_ID="1">Green Chili Salsa</data>
<data font_ID="2">Chopped Tomatoes</data>
<data font_ID="3">1 16-ounce Can</data>
<data font_ID="2">Chopped Green Chile</data>
<data font_ID="3">3 Tablespoons</data>
<data font_ID="2">Diced Onions</data>
<data font_ID="3">3 Tablespoons</data>
<data font_ID="2">Chopped Garlic</data>
<data font_ID="3">1 Tablespoon</data>
<data font_ID="2">Salt</data>
<data font_ID="3">1/4 Teaspoon</data>
<data font_ID="2">Lime Juice</data>
<data font_ID="3">1/2 Lime</data>
<data font_ID="2">Chopped Cilantro</data>
<data font_ID="3">1 Tablespoon</data>
<data font_ID="4">In a medium bowl, mix together all ingredients.</data>
<data font_ID="5">Use as a dip, or over grilled vegetables, or toss with bowtie noodles for an appetizer course.</data>
<data font_ID="6">www.</data>
<data font_ID="7">VeggieLife</data>
<data font_ID="6">.com</data>
<data font_ID="8">November 1997</data>

</document>
```

802

第 8 圖

804



第 9 圖

```

<?xml version="1.0" standalone="yes"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:template match="/">
    <document>
      <xsl:apply-templates/>
    </document>
  </xsl:template>
  <xsl:template match="document">
    <recipe title="{data[@font_ID='1']}">
      <xsl:for-each select="data[@font_ID='2']">
        <ingredient amount="{following-sibling::data[@font_ID='3']}">
          <xsl:value-of select="."/>
        </ingredient>
      </xsl:for-each>
      <xsl:for-each select="data[@font_ID='4']">
        <procedure step="{position()}">
          <xsl:value-of select="."/>
        </procedure>
      </xsl:for-each>
      <presentation>
        <xsl:value-of select="data[@font_ID='5']" />
      </presentation>
      <xsl:for-each select="data[@font_ID='7']">
        <originate name="{.}" "website="{preceding-sibling::data[@font_ID='6']}">{.}</following-
          sibling::data[attribute::font_ID='6']"> "date="{following-sibling::data[@font_ID='8']}" />
      </xsl:for-each>
    </recipe>
  </xsl:template>
</xsl:stylesheet>

```

1000